

## IN THE SPECIFICATION

After Paragraph [14], insert the following text:

The invention therefore provides a programmable machine for measuring time based on linear or non-linear programmable functions, comprising a data entry unit to input parameters that describe mathematical positive, monotonic functions, an arithmetic logical computation unit that performs operations on numeric data, by computing the programmed function, using time as the independent variable, a memory for storing numeric parameters for functions, counters of timer events, sections, results of function computation and mode selection, and a clock that generates equally spaced impulses, that are fed to the arithmetic logical unit for transformation.

Please amend paragraph [26] as follows:

[0026] A programmable machine for measuring time for exam events such as the SAT or PSAT with a purpose of improving student performance, comprising Display, data entry unit having Button pad keyboard comprising mode entry selection buttons, section selection buttons, option selection buttons, numeric buttons, Electronic memory that stores running parameters for each section of time, Logical Computation Unit that converts equally spaced time impulses from the clock to question counter increases based on a programmable function, Clock that generates equally spaced impulses, and Start/Stop button that switches mode from input mode to running mode.

Please cancel paragraph [35], and text between paragraphs [34] and [35], which reads:

“BACKGROUND OF THE INVENTION”

Please amend paragraphs 28-33:

[0028] FIG. ~~40~~ 1 is a plan view of the invention, with a physical appearance similar to a calculator, in setting mode.

[0029] FIG. ~~45~~ 2 is a plan view of the invention, with a physical appearance similar to a calculator, in running mode.

[0030] FIG. ~~20~~ 3 is a plan view of the invention, with a physical appearance similar to a wristwatch, in setting mode.

[0031] FIG. ~~25~~ 4 is a plan view of the invention, with a physical appearance similar to a wristwatch, in running mode.

[0032] FIG. ~~30~~ 5 is a schematic diagram illustrating the data flow between the elements of the invention.

[0033] FIG. ~~40~~ 6 is a flow chart of the operations that are performed by the logical unit.

Please amend paragraphs [43]-[55] as follows

[0043] Referring to FIG. ~~10~~ 1, a portable electronic timer of the type that may employ the features of this invention is shown in pictorial form. Its embodiment is similar to a portable electronic calculator. This particular embodiment of the invention is to be used in timing standardized tests, such as the SAT and the PSAT. These tests have a pre-determined number of sections. For each section, there are several groups of questions. Each group of questions is comprised by easy questions, medium difficulty questions and hard questions, in that order. It is difficult for the student taking these standardized tests to pace the effort, since more time is needed for the hard questions than for the easy ones. Some students, who very rarely are able to answer hard questions, may forego answering all or some of the hard questions and spend time for the easy questions and more time for the medium difficulty questions, in order to maximize their score. This electronic timer will show them what question they are supposed to be answering at any particular time. This information will allow them to think more about the question, guess, or skip the question depending of their being ahead or behind the question displayed by the timer.

[0044] The electronic timer 101 comprises a liquid crystal display 102, a data entry unit having button keyboard pad which comprises a questions set button 104, a time set button 105, a timer reset button 106, a section reset button 107, an SAT presetting button 108, a PSAT presetting button 109, two personal presetting memory buttons 110 and 111, two section setting buttons 112 and 113, numeric keypad with keys 0-9 103 ~~120~~, a clear entry button 114, an alarm setting button 115 ~~116~~, a timer mode button 116 ~~117~~ a Start/Stop button 117 ~~118~~ and an On/Off button 118 ~~119~~.

[0045] The operation of the device is as follows:

[0046] The user will turn the device on from the On/Off button 118. The display will show:

Section 1		PERS1
Time 0	Reset No	Timer Down
Questions 0	Reset No	Alarm Off

[0047] The user will press Time (button 105 ~~5~~). The display will blink the Time setting. The user will enter the time, in minutes allocated for this section. Setting the time in minutes is

particular to this application. Different applications may use different time measurements. Then the user will press Questions (button 104 -4-). The display will blink the Question setting. The user will set the number of questions for that section.

[0048] Then the user will decide if the timer will be reset at the beginning of this section by pressing the Time Reset button 106. This button toggles the Time Reset setting for the section. Then the user will decide if the question counter will be reset at the beginning of this section by pressing the Que. Reset button 107. This button toggles the Questions Reset setting for the section. When the timer is in the running mode, at the beginning of a section that indicates Time Reset, the displayed timer will start again from 0. This feature allows the user to time independently test sections, each of which is constituted of several timer sections. Each timer section corresponds to a sub-group of questions of the test section. When the timer is in the running mode, at the beginning of a section that indicates Question Reset, the displayed question starts again at 1. This feature allows the user to see to question counting independently for each test section.

[0049] Then the user will use the Section UP and DOWN ~~^~~ button buttons to advance to the next section and set it in a similar manner.

[0050] At any time, the user can select the Timer Mode by pressing on the Timer Mode button. If the timer mode is set to UP, the time displayed will go from 0 to the total for all the sections that do not require a timer reset. If the timer mode is set to DOWN, the time displayed is the sum of the specified time for all the sections up to the next section programmed for Time Reset. The time goes down from there and reaches 0 at the end that group of sections. This allows the user to know how much time is left for a certain test section.

[0051] Sections that have the Time set to 0 will be skipped, independently of setting of the Questions. Sections that have the Time set to a non-zero value, but the questions set to a 0 value, will consume the time specified, without increasing or modifying the Questions number. They are to be used to program the pauses in the test.

[0052] After all the desired sections are programmed, the user can select, using the Section UP ~~^~~ and Section DOWN \* buttons 112 and 113, the section from which to start the timer, then the user can press the Start/Stop button 117. This will start the timer and the screen will change to display the question;

[0053] This particular embodiment of the solution uses linear time functions. The slope

of the linear function is determined by the duration of the section and the programmed number of questions per section. The timer section of the programmed prototype provides an example of how the logical unit may be built.

[0054] The following is Visual Basic commented code, detailing the functionality of the logical unit. The logical unit will execute by hardware the same or similar logic functionality as described by this software. This prototype does not support the TimerMode DOWN. All other limitations of the prototype compared with the described electronic timer from FIG. 40 1 are described in the code comments.

```
6-Initialization-setting-the-counters. Unlike the described device, the prototype always starts
with section 1 and has only 6 sections. QuestionCounter = 1 DisplayQuestionCounter = 0
DisplayQuestionCounterOffset = 0 SectionCounter = 0 CurrentSeconds = 0 CurrentMinutes = 0
'Read the current computer time, the clock time StartTimer = Timer StartSectionTimer =
StartTimer 'display the section counter, the question counter and the starting time. If
(QuestionArray(SectionCounter) = 0) Then Text1.Text = "" Else Text1.Text = QuestionCounter
End If Text3.Text = SectionCounter + 1 Text4.Text = "00:00" 'Clock Event the clock is
programmed to create events much more often than the minimal time unit programmable by the
user 'In this case, the clock event is created every 10 ms, while the minimal time unit
programmable by the user is 1 minute. 'display time in section, first collect the current time from
the clock EndTime = Timer 'total time from the beginning of the group of sections, until timer
reset TotalTime = EndTime - StartTimer 'total time from the beginning of the current section
TotalSectionTime = EndTime - StartSectionTimer 'If the specified time is 0, skip the section; if it
is the last section, stop the running mode. 'otherwise compute what question from the current
section should be displayed. If TimeArray(SectionCounter) <= 0 Then SectionCounter =
SectionCounter + 1 StartSectionTime = EndTime 'set to current time If SectionCounter >=
MAX_SECTIONS Then Beep MsgBox ("End of Sequence") Unload Form1 End If Exit Sub
Else 'increase section time QuestionCounter = TimeTransform(TimeArray(SectionCounter),
QuestionArray(SectionCounter), TotalSectionTime) End If 'if the number of questions
programmed for the current section is 0, then the QuestionCounter is kept to 0 for the section
and the TotalSectionTime is compared to the programmed time for the section. The 60
multiplier is used to transform the time from minutes to seconds, to match the TotalSectionTime
units. When the time expires, the timer will move to the next section. If
(QuestionArray(SectionCounter) = 0) Then QuestionCounter = 0 If TotalSectionTime > 60 *
TimeArray(SectionCounter) Then SectionCounter = SectionCounter + 1 StartSectionTime =
EndTime 'set to current time If SectionCounter >= MAX_SECTIONS Then Beep MsgBox
("End of Sequence") Unload Form1 End If Exit Sub End If End If 'if the question number for this
section is higher than the number of questions requested for the section, it is time for a section
increase. the timer moves to the next section. The section start timer is set to the current time.
The timer displayed and the question displayed are reset, if so programmed. the
DisplayQuestionCounterOffset keeps track of the questions in previous sections, that need to be
added to the questions in the current section, before a question number is displayed. The
```

```

QuestionCounter, 'which stores the number of the question to be displayed from the current
'section is set back to 1. If QuestionCounter > QuestionArray(SectionCounter) Then
SectionCounter = SectionCounter + 1 StartSectionTimer = EndTime 'set to current time If
SectionCounter >= MAX_SECTIONS Then Beep MsgBox ("End of Sequence") Unload Form1
Exit Sub End If If TimerReset(SectionCounter) = 1 Then StartTimer = StartSectionTimer End If
If QuestionReset(SectionCounter) = 1 Then DisplayQuestionCounterOffset = 0 Else
DisplayQuestionCounterOffset = QuestionArray(SectionCounter - 1) + .sub.
DisplayQuestionCounterOffset End If QuestionCounter = 1 End If 'data is ready for display.
DisplayQuestionCounter is the question number 'displayed on the screen.
DisplayQuestionCounter = DisplayQuestionCounterOffset + QuestionCounter 'display the
question number, unless the current section is a pause 'section, with no questions programmed
for it. If (QuestionArray(SectionCounter) = 0) Then Text1.Text = "" Else Text1.Text =
DisplayQuestionCounter End If 'Total time keeps track of the time since the last Time Reset. The
number 'is converted into seconds and minutes and displayed. The prototype does 'not keep track
of groups of sections longer than an hour. The section 'counter is displayed. CurrentSeconds =
Int(TotalTime) Mod 60 CurrentMinutes = Int(Int(TotalTime) / 60) If (CurrentSeconds < 10)
Then strCurrentSeconds = "0" + CStr(CurrentSeconds) Else strCurrentSeconds =
CStr(CurrentSeconds) End If If (CurrentMinutes < 10) Then strCurrentMinutes = "0" +
CStr(CurrentMinutes) Else strCurrentMinutes = CStr(CurrentMinutes) End If Text4.Text
strCurrentMinutes + ":" + strCurrentSeconds Text3.Text SectionCounter + 1 'The
TimeTransform function is in this case linear. 'The number of Ticks is the number of questions in
this case. Function TimeTransform(iTime As Integer, iNumberOfTicks As Integer,
TotalSectionTime As Variant) As Integer Dim LinearTime As Double If iNumberOfTicks > 0
Then LinearTime = Int(iTime * 60 / iNumberOfTicks) Else LinearTime = iTime End If
TimeTransform = Int(TotalSectionTime / LinearTime) + 1 End Function

```

[0055] Referring to FIG. 3 20, a portable electronic timer of the type that may employ the features of this invention is shown in pictorial form. Its embodiment is similar to a wristwatch. The electronic timer 201 comprises a liquid crystal display 202, a button keyboard pad which comprises a setting button 203, an exam setting button 204, an Up/Down button 205 and a start/stop button 206.



Please insert the attached computer software code appendix after paragraph [63]

Streja-201

- 15 -

09/815,910

## COMPUTER PROGRAM CODE APPENDIX

'Initialization setting the counters. Unlike the described device, the prototype  
'always starts with section 1 and has only 6 sections.

```
QuestionCounter = 1
DisplayQuestionCounter = 0
DisplayQuestionCounterOffset = 0
SectionCounter = 0
CurrentSeconds = 0
CurrentMinutes = 0
```

```
'Read the current computer time, the clock time
StartTimer = Timer
StartSectionTimer = StartTimer
```

'display the section counter, the question counter and the starting time.

```
If (QuestionArray(SectionCounter) = 0) Then
    Text1.Text = ""
Else
    Text1.Text = QuestionCounter
End If
Text3.Text = SectionCounter + 1
Text4.Text = "00:00"
```

'Clock Event - the clock is programmed to create events much more  
'often than the minimal time unit programmable by the user  
'In this case, the clock event is created every 10 ms, while the minimal  
'time unit programmable by the user is 1 minute.  
'display time in section, first collect the current time from the clock

```
EndTime = Timer
```

'total time from the beginning of the group of sections, until timer reset

```
TotalTime = EndTime - StartTimer
```

'total time from the beginning of the current section

```
TotalSectionTime = EndTime - StartSectionTimer
```

'If the specified time is 0, skip the section; if it is the last section,  
'stop the running mode, otherwise compute what question from the current  
'section should be displayed.

```

If TimeArray(SectionCounter) <= 0 Then
    SectionCounter = SectionCounter + 1
    StartSectionTime = EndTime 'set to current time
    If SectionCounter >= MAX_SECTIONS Then
        Beep
        MsgBox ("End of Sequence")
        Unload Form1
    End If
    Exit Sub
Else
    'increase section time
    QuestionCounter =
TimeTransform(TimeArray(SectionCounter),
    QuestionArray(SectionCounter), TotalSectionTime)
End If

```

'if the number of questions programmed for the current section is 0, then  
'the QuestionCounter is kept to 0 for the section and the TotalSectionTime is  
'compared to the programmed time for the section. The 60 multiplier is used to  
'transform the time from minutes to seconds, to match the TotalSectionTime  
'units. When the time expires, the timer will move to the next section.

```

If (QuestionArray(SectionCounter) = 0) Then
    QuestionCounter = 0
    If TotalSectionTime > 60 * TimeArray(SectionCounter)
Then
        SectionCounter = SectionCounter + 1
        StartSectionTime = EndTime ' set to current time
        If SectionCounter >= MAX_SECTIONS Then
            Beep
            MsgBox ("End of Sequence")
            Unload Form1
        End If
        Exit Sub
    End If
End If

```

'if the question number for this section is higher than the number of  
'questions requested for the section, it is time for a section increase.  
'the timer moves to the next section. The section start timer is set to  
'the current time. The timer displayed and the question displayed are  
'reset, if so programmed.  
'the DisplayQuestionCounterOffset keeps track of the questions in

'previous sections, that need to be added to the questions in the current  
'section, before a question number is displayed. The QuestionCounter, which  
'stores the number of the question to be displayed from the current section  
'is set back to 1.

```
If QuestionCounter > QuestionArray(SectionCounter) Then
    SectionCounter = SectionCounter + 1
    StartSectionTimer = EndTime ' set to current time
    If SectionCounter >= MAX_SECTIONS Then
        Beep
        MsgBox ("End of Sequence")
        Unload Form1
        Exit Sub
    End If
    If TimerReset(SectionCounter) = 1 Then
        StartTimer = StartSectionTimer
    End If
    If QuestionReset(SectionCounter) = 1 Then
        DisplayQuestionCounterOffset = 0
    Else
        DisplayQuestionCounterOffset =
        QuestionArray(SectionCounter - 1) +
        DisplayQuestionCounterOffset
    End If
    QuestionCounter = 1
End If
```

'data is ready for display. DisplayQuestionCounter is the question number  
'displayed on the screen.

```
DisplayQuestionCounter = DisplayQuestionCounterOffset +
    QuestionCounter
```

'display the question number, unless the current section is a pause  
'section, with no questions programmed for it.

```
If (QuestionArray(SectionCounter) = 0) Then
    Text1.Text = ""
Else
    Text1.Text = DisplayQuestionCounter
End If
```

'Total time keeps track of the time since the last Time Reset. The number  
'is converted into seconds and minutes and displayed. The prototype does  
'not keep track of groups of sections longer than an hour. The section  
'counter is displayed.

```
CurrentSeconds = Int(TotalTime) Mod 60
CurrentMinutes = Int(Int(TotalTime) / 60)

If (CurrentSeconds < 10) Then
    strCurrentSeconds = "0" + CStr(CurrentSeconds)
Else
    strCurrentSeconds = CStr(CurrentSeconds)
End If
If (CurrentMinutes < 10) Then
    strCurrentMinutes = "0" + CStr(CurrentMinutes)
Else
    strCurrentMinutes = CStr(CurrentMinutes)
End If
Text4.Text = strCurrentMinutes + ":" + strCurrentSeconds
Text3.Text = SectionCounter + 1
```

'The TimeTransform function is in this case linear.  
'The number of Ticks is the number of questions in this case.  
'It can be changed to any functional transformation.

```
Function TimeTransform(iTime As Integer, iNumberOfTicks As
Integer, TotalSectionTime As Variant) As Integer
    Dim LinearTime As Double

    If iNumberOfTicks > 0 Then
        LinearTime = Int(iTime * 60 / iNumberOfTicks)
    Else
        LinearTime = iTime
    End If

    TimeTransform = Int(TotalSectionTime / LinearTime) + 1
End Function
```